A RAND NOTE

TOWARD AN EXPERT AID FOR TACTICAL AIR TARGETING

Monti Callero, Daniel Gorlin,
Frederick Hayes-Roth, Lewis Jamison

January 1981

N-1645-ARPA

Prepared For

The Defense Advanced Research Projects Agency

Rand
SANTA MONICA, CA. 90406

The research described in this report was sponsored by the Defense Advanced Research Projects Agency under ARPA Order No. 3460/3473, Contract No. MDA903-78-C-0029, Information Processing Techniques.

The Rand Publications Series: The Report is the principal publication documenting and transmitting Rand's major research findings and final research results. The Rand Note reports other outputs of sponsored research for general distribution. Publications of The Rand Corporation do not necessarily reflect the opinions or policies of the sponsors of Rand research.

Published by The Rand Corporation

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

RAND/

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
N-1645-ARPA ✓

2. GOVT ACCESSION NO.
AD-A097720

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
(6) Toward an Expert Aid for Tactical Air Targeting.

5. TYPE OF REPORT & PERIOD COVERED
(9) Interim rept.,

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
(10) Monti/Callero, Daniel/Gorlin, Frederick/Hayes-Roth, Lewis/Jamison

8. CONTRACT OR GRANT NUMBER(s)
(15) MDA903-78-C-0029 ✓

9. PERFORMING ORGANIZATION NAME AND ADDRESS
The Rand Corporation ✓
1700 Main Street
Santa Monica, CA. 90406

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
(12) 61

11. CONTROLLING OFFICE NAME AND ADDRESS
Defense Advanced Research Projects Agency
Depart of Defense
Arlington, VA. 22209

12. REPORT DATE
(11) Jan~~uary~~ 1981

13. NUMBER OF PAGES
49

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
(14) RAND/N-1645-ARPA

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release: Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

No Restrictions

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Artificial Intelligence
Heuristic Methods
Computer Programming Language
Tactical Intelligence
Target Intelligence

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

See Reverse Side

UNCLASSIFIED 296600 slt

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

This Note describes our initial efforts to apply recent advances in knowledge engineering to the domain of tactical air targeting. Tactical targeting is a critical function in war requiring many complex, heuristic, and time stressed decisions by the taregeteer. A knowledge engineering approach to providing and aid for this process suits the domain. First, knowledge employed by targeteers does not lend it self to straightforward computer implementation. Second, no standard approach exists for targeting. Third, by "engineering" targeteers' knowledge, a basis is provided for experimentation and reformulation of targeting concepts and practices. Finally, the operational environment requires effective human interaction and ready program modification. Knowledge engineering has greater potential to meet these needs than other programming approaches. Our research tests the hypotheses that an expert system would improve tactical targeting and that knowledge engineering can be extended to meet the task. The paper describes the technical and targeting environments, early project experiments, the latest targeting program, TATR, written in ROSIE-I, and our current approach using ROSIE-II.

A RAND NOTE

TOWARD AN EXPERT AID FOR TACTICAL AIR TARGETING

Monti Callero, Daniel Gorlin,
Frederick Hayes-Roth, Lewis Jamison

January 1981

N-1645-ARPA

Prepared For

The Defense Advanced Research Projects Agency

Rand
SANTA MONICA, CA. 90406

PREFACE

Under the support of the Information Processing Techniques Office of the Defense Advanced Research Projects Agency, Rand has been investigating the possibility of applying new technology in the field of artificial intelligence to the problem of Air Force tactical planning. This research has focused on the possibility of using the tools and techniques of knowledge engineering to construct an intelligent assistant "expert system" for tactical targeting. This Note explains the tactical targeting problem and describes several preliminary systems developed by Rand for coping with the targeting problem. The work was conducted under ARPA Contract No. MDA903-78-C-0029.

SUMMARY

Recent advances in knowledge engineering, a branch of artificial intelligence, have led to the development of numerous "expert systems." These systems achieve performance in difficult problem domains at a level comparable to that of human experts. This Note describes the initial efforts undertaken at Rand to apply this technology to Air Force tactical planning problems.

The function of tactical targeting is to evaluate the importance of various enemy targets, to select targets to attack, and to determine suitable weapons and tactics to employ. This is a critical function in war. It requires many complex, heuristic, and time-stressed decisions on the part of the human targeteer.

The development of a computer-based intelligent assistant for the targeteer presents many difficult problems. First, the knowledge that targeteers employ does not lend itself to straightforward computer implementation. It is often difficult to express precisely or to incorporate in a problem-solving system. Second, no standard approach exists for targeting. In current practice, different targeteers employ significantly different methods and consider a variety of different issues. Third, by "engineering" the targeteers' knowledge we provide a better basis for experimentation and reformulation than is normally possible with knowledge that resides exclusively in human heads. This situation gives rise to the common practice of experimental implementation and iterative reimplementation that normally occurs in knowledge-engineering applications. Fourth, in the tactical targeting

application, knowledge is less rigorous than it is in other application domains to which this technology has been previously applied. The needs for human interaction and interface design are greater than in many other applications.

This Note describes the basic components of the targeting problem and several initial preprototype systems that we have developed for assisting the targeteer. Each of these systems addresses various aspects of the overall targeting problem and uses a variety of tools and techniques for the task. These systems provided bases for constructive reviews which, in turn, led to revised design and development concepts for the targeting aids.

The Tactical Air Targeting Recommender (TATR), the latest of these experimental systems, is explained in considerable detail. We illustrate the knowledge representations and algorithms that TATR employs. These illustrations include samples of the actual computer code written in the ROSIE-I language. Subsequent development of the TATR system is under way using the newest version (II) of the ROSIE language. Many of the newest concepts in ROSIE for improved readability, simplicity, and programming efficiency have arisen in response to our critique of ROSIE-I as a language for tactical application of knowledge-engineering techniques.

ACKNOWLEDGMENTS

The authors wish to acknowledge the valuable contributions of Stephanie Cammarata and Stan Rosenschein to the research efforts reported in this Note. Stephanie wrote many early routines in INTERLISP and wrote the damage expectancy calculation routines for the Tactical Air Target Recommender (TATR). Stan led the project during its first year, formulated the initial approaches, and wrote the Tactical Air Selection System (TASS). Both efforts were essential to our progress.

CONTENTS

FIGURES

I. INTRODUCTION

Over the past two years, Rand has been engaged in a research effort to apply recent technological advances in knowledge engineering to Air Force tactical planning. This Note describes the research, the interim results achieved to date, and the direction of future efforts.

Tactical air planning requires searches among thousands of potential targets to identify critical and vulnerable elements whose destruction would support prescribed military objectives. Currently, target selection results from human judgments. These judgments integrate information about the enemy's force posture and capabilities as well as information on friendly capabilities to conduct tactical air operations against enemy targets. The judgment process does not now employ automated support. Our research hypothesizes, and we strongly believe, that carefully designed automated aids for the tactical planner can improve these judgments and thereby should directly improve the effectiveness of U.S. tactical air operations.

Knowledge engineering provides problem-solving approaches especially designed for predominantly human judgment tasks such as this one. However, knowledge engineering has not yet been applied to a military operational decisionmaking environment. These dynamic environments often require a planner to make time-constrained critical choices among many complex alternatives with uncertain outcomes. At issue is whether or not this new technology is sufficiently developed to handle such realistic and complex problems.

Our n has attempted to determine if state-of-the-art know' gineering techniques can augment complex military nmaking. We have addressed this objective by developing a totype "expert system" to assist in a major function within the tactical air planning process, <u>the selection and prioritization of targets</u>.

The development process for an operational prototype has three parts.

1. Select a manageable subset of the targeting domain and acquire a representative knowledge base--the targeting heuristics and data files.
2. Build a "basic" prototype system to determine the feasibility of satisfying the selected targeting requirements using knowledge engineering.
3. Evaluate and evolve the prototype system in conjunction with Air Force targeteers, seeking to achieve an operationally acceptable level of performance with regard to quality of output, responsiveness, human interface, and system adaptability.

To date we have expended our efforts only on the first two parts, experimenting with problem domains and knowledge-engineering systems; we have not yet achieved sufficiently satisfactory results to proceed with the third part.

In the remainder of this section, we discuss knowledge engineering and tactical air planning. In Sec. II, we describe our research approach. Sections III and IV are overviews of our early efforts and our most recent major research product, the Tactical Air Target Recommender (TATR), a program written in the Rand-developed ROSIE-I programming language for knowledge-based systems. In Sec V, [illegible] is the direction of future research.

KNOWLEDGE ENGINEERING

Recent advances in artificial intelligence (AI) have demonstrated the primary importance of domain knowledge in achieving expert-level problem-solving performance. In contrast with earlier AI studies, these later application programs use considerably more world knowledge and common sense to simulate the reasoning of a human expert. This focus on representing and applying expert knowledge has given rise to a subspecialty called "knowledge engineering."

From the experience gained in this field, a few tenets have emerged that guide knowledge-engineering applications. The first is that a successful project team needs to include both computer experts and problem-domain experts. These two cooperate in the development of a knowledge base specific to the problem domain. This knowledge base incoporates domain data, general facts about relationships that occur in the domain, and problem-solving heuristics. The heuristics reflect the human expert's rules-of-thumb for reasoning about problems in this domain. Applying these elements of knowledge to a specific task requires some sort of problem-solving mechanism. Such a mechanism, often called an "inference engine" or a "deductive procedure," determines which elements of knowledge are relevant to the problem at hand and chooses a sequence of inferences to perform.

Knowledge engineering often requires many iterations of system implementation. Unlike more conventional applications, the tasks undertaken by knowledge engineers frequently lack clear-cut formulations and correspondingly straightforward solutions. In short, the knowledge that human experts possess is often difficult to articulate because it may be incomplete, fuzzy, or inconsistent. Translating such knowledge into computer programs produces precise and rigorous intepretations which must be reworked repeatedly to emulate the intuitive knowledge underlying human judgment. During the iterative process of formulating, implementing, and testing knowledge, we often gain a deeper understanding of the problem domain. Such improved understanding then stimulates changes in the knowledge base to reflect new perceptions. For these reasons, knowledge engineering generally requires an evolutionary approach to system development.

Nearly all previous applications of knowledge engineering have focused on non-military problems. The most successful applications have been in scientific areas having extensive amounts of rigorous scientific knowledge, such as chemistry and some narrow areas of medicine. But military planning, which is our chosen application domain, also presents many opportunities and needs for knowledge-engineered systems. Many difficult decisions must be made with the aid of data, facts, and heuristics. These decisions frequently require capabilities or resources that exceed the capacities of the human decisionmakers. Thus, these humans could benefit from intelligent-systems assistance in planning. In fact, we see the greatest immediate need for knowledge engineering in the area of developing tools to assist human planners and analysts. This focus places a greater emphasis on the need for human-machine interaction than previous knowledge-engineering applications have faced.

TACTICAL AIR PLANNING

In wartime, the tactical air planning process determines the intended operational use of tactical air resources in a future operational time period (historically, the next day) and prepares the necessary orders and instructions for operational units (e.g., fighter wings) to execute the planned missions. Figure 1 shows the process cycle containing four major steps--target file generation, targeting, force application, and Air Tasking Order preparation.

Target File Generation

Collection of intelligence data (information about the enemy) takes place continuously, beginning well in advance of a war and increasing in pace and focus after hostilities commence. Many sources gather raw data, using a wide variety of techniques ranging from purely human efforts to applications of some of our most advanced technology. Intelligence analysts reduce the raw data to identify and classify enemy resources and force elements and then construct a target base composed of large data files of potential targets. During the course of conflict, the status of the potential targets can change rapidly as a result of actions against them or of the enemy's own operations; hence, the target base undergoes frequent modification as new intelligence is reported and analyzed.

This target base provides the main source of information on the enemy for the tactical planning process. For each potential target, the

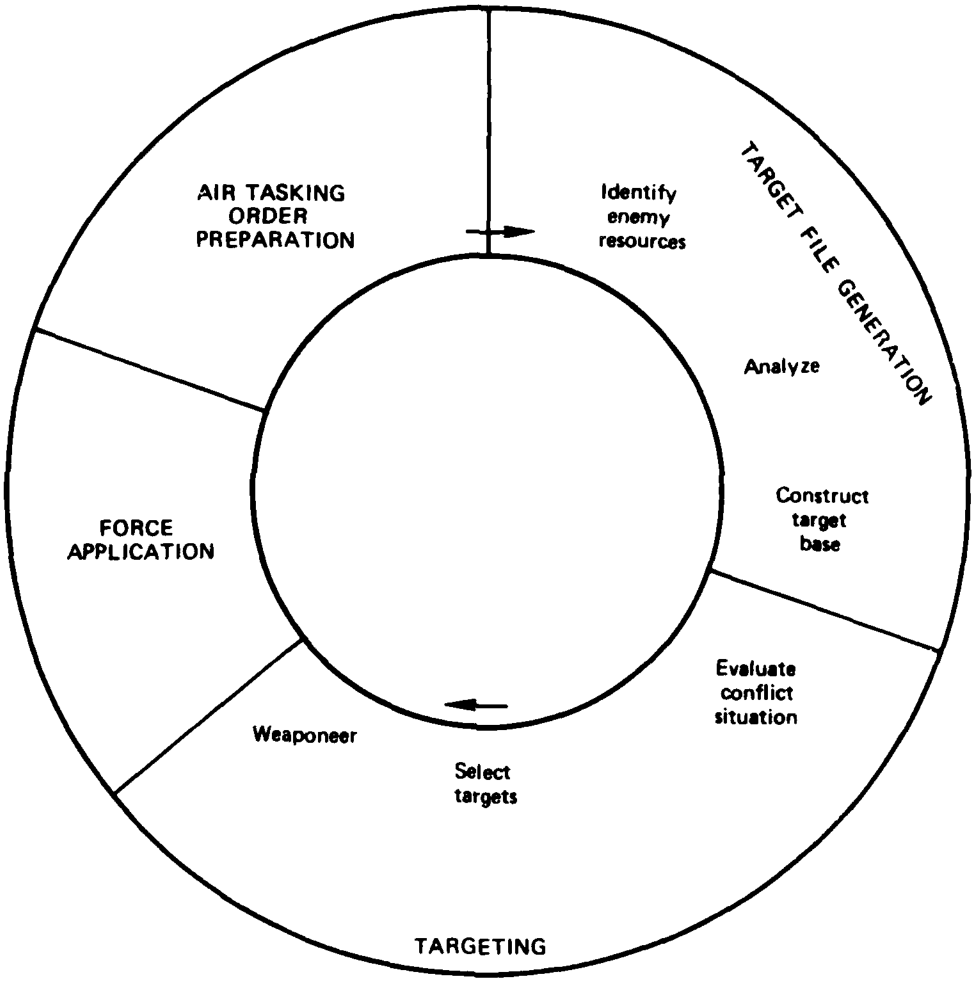

Fig. 1--Tactical air planning cycle

information can include target type, location, organizational linkages, supporting elements, recent movements, and estimates of capabilities. For installations, such as airfields, similar data are included for force elements (e.g., aircraft), support elements (e.g., maintenance), and facilities (e.g., petroleum storage) located at the installation.

In the USAF Tactical Air Control Center (TACC), where the tactical air planning process takes place, the target base is partially automated by the Data Communication, Storage and Retrieval System (DC/SR). The remainder of the target base resides in hardcopy text, maps, and photographs.

Targeting

The targeting function selects from the target base specific targets for attack and identifies weapon systems that can attack the targets and achieve desired damage expectancies. The identification of weapons is called "weaponeering." Targeting involves three overlapping and iterative activities: evaluation of targets in the target base to assess their military value and relevance; selection of a candidate subset of targets and determination of the effects desired against them; and weaponeering to determine both ability to achieve the desired effects and expected resource costs

Selection of a subset of targets from the many thousands of candidates in the target base must be based on the significance, accessibility, and vulnerability of targets; objectives and strategies set forth in apportionment instructions and other guidance documents; rules of engagement; principles of air warfare; and tactics. Effects expected to be achieved against selected targets are determined from target analysis information such as vulnerability, perishability, utility value, relationship to other targets, location and mobility, and validation status. Weaponeering calculations array damage criteria and weapons effects against forces, weapons, fuzing, and delivery tactics to provide numbers of aircraft with specified munitions required to attain desired expected damage levels on each target. Also, information on enemy defenses is analyzed, and a defense suppression target list is prepared for each target.

The targeting process results in a prioritized list of targets for attack during the following day, based on all of the considerations and information accumulated from the above activities. The list is passed to the next step in the planning process, along with weaponeering data and defense suppression targets.

Force Application

Force application produces a plan matching friendly air resources and enemy targets. Inputs to the process include the prioritized target list, defense suppression target list, and weaponeering data prepared in the targeting process; threat estimates; availability and capability of friendly forces; weather; and combat objectives, strategies, and tactics. The goal is to generate an assignment of available forces to the target set in such a way as to achieve the best possible tradeoff between results and cost. Forces are generally assigned in strike packages which, may include defense suppression aircraft, fighter escorts, electronic countermeasure aircraft, and reconnaissance aircraft, in addition to the aircraft actually attacking the target, and which may possibly require overflight coordination with friendly ground fire-support elements.

The plan specifies the units that are to fly the missions, the types of aircraft, the munitions to be carried, the controlling agencies (e.g., ground radar sites) to be utilized going to and from the target, and the timing of the critical points in the mission, such as rendezvous with tanker and escort aircraft and time over the target.

Air Tasking Order Preparation

The final step in the planning process formats the agreed-upon plan as an Air Tasking Order (ATO) that is promulgated to all appropriate organizations. It directs them to perform the attacks as specified. Currently in the tactical air control center, handwritten plan information provided by the force application section is punched onto paper tape or cards for distribution through the teletype communications net. This time-consuming, laborious, error-prone procedure is expected to be replaced within the next year by an automated ATO preparation and monitoring capability as part of the first increment of the Computer Aided Force Management System (CAFMS).

II. RESEARCH APPROACH

OBJECTIVE

The tactical planning process is characterized by time-constrained application of human judgment to complex problems at every step. These decisions are made by Air Force officers with a variety of experience and backgrounds. Initially, they cannot be expected to have broad experience in tactical planning, and certainly nothing approaching expertise. This fact, together with the inherent complexity of tactical planning, indicates the need for sophisticated, automated aids. Moreover, the unpredictability of warfare will preclude rigidity in concepts, procedures, and decision processes in whatever automated aids are provided.

It is our belief that an aid which can accumulate knowledge, consider heuristics, adapt to the user as well as the situation, and communicate easily with the user would significantly improve the force employment process. We further believe that continuing advances in knowledge engineering tools and techniques have brought knowledge engineering to a point where it can serve as the basis for such an aid. The purpose of our research is to test these hypotheses.

We have chosen to focus on the targeting step because it is separable both notionally and in practice, because it contains sufficient elements to fully challenge current knowledge-engineering capabilities, and because it would fulfill a real need for the targeteer if successful. Our approach has been to build a workable prototype with operational capability that contains important improvements to the targeting process.

As mentioned earlier, the development process of a prototype has three parts--the selection of a specific targeting domain and the acquisition of the knowledge base relevant to that domain; the building of a prototype that has basic features to test the feasibility of applying knowledge-engineering techniques; and, if feasibility is demonstrated, the evolution of the prototype toward an operationally acceptable capability in terms of quality of output, responsiveness, human interfaces, and adaptability. In our work thus far, we have concentrated on the air base attack domain, with an excursion into ground-force attack in our most recent effort (reported in Sec. IV). We have acquired necessary reoresentative knowledge and have built prototype systems with rudimentary targeting capabilities. However, none of these systems has shown sufficient potential to permit continuation into the evolution phase of the development process. Rather, they have demonstrated the need for the more capable knowledge-engineering tools that have been under development.[1] In the remainder of this section we address knowledge acquisition and knowledge-engineering system building.

KNOWLEDGE ACQUISITION

Two categories of knowledge must be acquired:

[1] Daniel Gorlin, Frederick Hayes-Roth, Stanley Rosenschein, Henry Sowizral, and D. A. Waterman, Rationale and Motivation for ROSIE, The Rand Corporation, N-1648-ARPA (forthcoming).

1. Knowledge held by humans and used by them to process information about the conflict situation in order to make decisions about the use of tactical air resources.
2. Information about the conflict environment that is known (or at least reported) and is available to the decisionmakers (generally stored in data files or data bases).

The human knowledge to be used in a knowledge-engineering system must be acquired directly from persons expert (or at least very knowledgeable) in performing the targeting tasks. The conflict environment information base must be provided by information-gathering systems external to the knowledge-engineering system, and the information must be adapted to a structure and format usable by the knowledge-engineering program.

Tactical Targeting Expertise

Air Force personnel who are trained and have some experience in tactical air targeting form the primary source of targeting knowledge for this project. This source is limited, however, since in the peacetime Air Force environment, these personnel do not function primarily in a tactical air targeting capacity. They are assigned to a variety of locations and jobs--staff positions, training classrooms, and tactical units--resulting in a very distributed knowledge source. Nevertheless, small pockets of targeteers do exist in the Tactical Control Wings, the C3I Complex at Hurlburt Field, the joint intelligence school at Lowry AFB, and the headquarters of tactical air forces (TAC, USAFE, PACAF). They constitute our primary knowledge sources.

We are mainly interested in large-scale, modern conflict, where the most targeting help would be needed and the greatest returns would be expected. Again, we are hampered by the lack of an experience base within the USAF in conducting air operations against sophisticated forces in this type of conflict. The lack of practical targeting expertise in the conflict environment of highest interest for this study necessitates the use of related experience (e.g., from the Southeast Asia/Vietnam conflict or military exercises), simple lore, and doctrine to project effective targeting techniques in that environment.

Iterative/Evolutionary Human-Knowledge Acquisition

The distributed nature of expert targeting sources and the general lack of expertise in fighting against modern forces in large-scale war have important implications for the utility and structure of the targeting aid as well as the knowledge-acquisition techniques. From the utility standpoint, the targeting aid itself provides a focal point and serves as the repository for the development and accumulation of (prewar) targeting concepts by the Air Force targeting community. If the distributed knowledge can be centered in a functional tool permitting experimentation, evaluation, and modification, the Air Force might have at least a reasonably adequate, well-considered targeting capability at the outbreak of war. On the other hand, the targeting aid will need to adapt to changing ideas about war-fighting that evolve in peacetime as well as during an actual war, since to a large degree we would have to learn to fight a future war as it unfolds. Hence, the structure of the targeting aid must permit rapid adaptation within the operational environment.

The main implication for targeting-knowledge acquisition is that it must be an ongoing process, even in an operational wartime environment. The knowledge must evolve over time through iterations of trial and evaluation, and the targeting aid itself must contribute to the process.

In this research, our approach to acquiring an initial set of targeting knowledge has been to identify highly qualified Air Force targeteers having some experience in tactical air targeting and to discuss their targeting techniques with them. After eliciting an initial set of targeting heuristics, we formalize and structure those heuristics and iteratively, with the targeteers, improve our interpretation and the conciseness and precision of their rules. After sufficient agreement, we implement the heuristics in a program which then becomes the primary vehicle for evolving further heuristics in concert with the entire community of targeteers.

Conflict Environment Information Bases

To provide a context for the elicitation of targeting heuristics and the generation of resultant target selections, we adopted the conflict information pertaining to the Tactical Air/Land Integrated Exercise (TALIE) developed by the Air Force and used extensively by the Air/Ground Operations School and others. It represents a (much scaled-down) European conflict between the Warsaw Pact and NATO. All the information is unclassified, which provides for research flexibility.

The TALIE data base contains over 300 major targets, including supply depots, highway junctions, highway bridges, railroad bridges, railroad yards, power plants, radar sites, surface-to-air missile sites, airfields, army barracks, and military headquarters. Additionally, approximately 300 more specific targets are located at the major targets (e.g., aircraft or fuel storage facilities on an airfield). To this data base, we have added 40 enemy combat divisions arrayed in a representative attack pattern.

To support the weaponeering function (the relating of expected target damage and specific attacks by various aircraft/munitions combinations) without using classified information and procedures, we developed a representative weaponeering procedure and generated an unclassified weaponeering data base which is sufficiently realistic for research purposes.[2]

The weaponeering data base consists of a damage-probability file and a survival-probability file. The damage-probability file has a set of entries for each target type. The sets contain an entry for each combination of weapon system (F-4, F-111, A-10), munitions load (Mk 82, Mk 83, Maverick, CBU, LGB), and delivery tactic (high angle, low angle, level) that is feasible for use against that target type. Each entry specifies the probability that the target will be destroyed if a single aircraft delivers the munitions against it, using the prescribed tactic. The survival-probability file has an entry for each feasible combination of aircraft type and delivery tactic, which specifies the probabilities that the aircraft will survive in high-threat and low-threat enemy enroute and target-area defense environments. All entries in these

[2] In an actual operational environment, automated weaponeering calculations can be accessed by interfacing with existing programs on the DC/SR.

files are constructive, i.e., generated without reference to actual data, to assure an unquestionably unclassified status. They are, however, credible in a general sense and can also be readily changed.

KNOWLEDGE-ENGINEERING SYSTEMS

To develop a knowledge-engineering system, we must make several initial choices and focus on a small part of the problem. Having selected some existing knowledge-engineering tool or "representation language," we implement a prototype system in that language. Such a system attempts to solve the chosen part of the problem within the framework provided by the knowledge-engineering tool. A variety of tools and languages exist from which such initial selections can be made. In this study, we have alternately used production systems, opportunistic planning, natural-language data base queries, and a variety of other frameworks for formulating tactical planning knowledge.

We have also used a variety of existing knowledge-engineering tools. Our first implementation was developed in Rand's RITA system on a minicomputer. At that time, RITA was the best available system for encoding knowledge in readable "if-then" rule forms. Our second system was developed in INTERLISP and was designed to support a potentially large and complex heuristic search of planning alternatives. This system provided capabilities for incorporating diverse programs that could cooperate by using different sources of knowledge. The third system we implemented was also written in INTERLISP, but it provided a friendly, English-like interface to a tactical planning data base. Our fourth system was developed in the first version of Rand's new rule-based programming language, ROSIE. This system incoporated heuristics for selecting and weaponeering ground-force and airfield targets.

Each of these initial systems investigated alternative and complementary design issues that our project must address. Each has added specificity to our understanding of the tactical planning problem in general and the qualities which an intelligent assistant for this domain should possess. In the next section, we review briefly these initial system-development efforts.

III. INITIAL EFFORTS

We have developed four initial prototypes of an "intelligent assistant" for tactical targeteers. Each of these prototypes investigated different portions of the problem domain and employed different formalizations and inference methods. These initial systems are described briefly below.

DEVELOPMENT STEPS

TATR

The Tactical Air Target Recommender (TATR) addressed the problem of choosing specific targets on an airfield to accomplish predetermined objectives for attacking the airfield. Based on heuristics provided by a user which establish preferential ordering of targets depending on the objectives to be accomplished, TATR develops a preferentially ordered list of strike recommendations--targets with weapon-system selection.

TATR is written in the RITA (Rule-directed Interactive Transaction Agent) programming language, which allows rules to be written in quasi-English and provides an operating environment in which these rules are interpreted. Because a user can readily understand and modify the RITA program, the RITA programming environment has been an excellent medium for demonstrating the use of rule-based systems and for soliciting information from experts.

Our experience with TATR supported several conclusions. The quasi-English formalism of RITA programs proved very attractive to military personnel, and the problem of air base attack was generally accepted as important and worthy of our efforts. However, the RITA language and its minicomputer environment were inadequate for the research-and-development phase of our knowledge-engineering project.

TAC PLANNER

Our second implementation effort, TAC PLANNER, addressed a more general architecture problem. The TAC PLANNER system provided mechanisms for incorporating diverse sources of knowledge in a cooperative problem-solving paradigm. Such an architecture has proved useful in other AI knowledge-engineering tasks, including speech understanding and signal interpretation. The advantage of such an architecture is that new sources of knowledge can be added incrementally and the overall control of problem-solving activities can be governed by a separate specialist program. If such systems have disadvantages, they arise from the lack of a restrictive syntax for representing knowledge. Without restricting the ways in which knowledge is represented, it is difficult to allow an end-user to read or modify the computer code or to provide general-purpose explanation facilities.

Our TAC PLANNER was developed in the INTERLISP programming language. In addition to providing for multiple sources of knowldge, TAC PLANNER also incorporated a rudimentary relational data base with dependency relations among data-base assertions. These dependencies supported automatic belief revision in response to data-base changes. Thus, when new data arrived from external sensors, TAC PLANNER would automatically propagate their effects to all affected beliefs and knowledge sources. This architecture provided the basic ingredients for a planner's assistant that could operate effectively in a situation with dynamically changing data.

As it turned out, we encountered severe obstacles in attempting to integrate our TAC PLANNER with actual Air Force data-base systems, because those systems do not currently support interaction with remote computers. This difficulty, coupled with the illegibility of INTERLISP code for military users, led us to move to a third system. However, we learned several useful lessons from our experience with TAC PLANNER. Dynamic updating of inferences required special mechanisms, but we find these achievable. Integration of diverse sources of knowledge seems desirable, and this requires a knowledge representation that modularizes the diverse specialties. Conclusions often depend on many distributed prior computations. The dependencies among inferences can be stored to aid explanation and dynamic updating. Finally, for interaction with military users, readable code seems more important than flexibility in coding capabilities; therefore, representing knowledge in INTERLISP seems a poor idea.

TASS

The third system we developed was called TASS (Tactical Air Selection System). Based on our previous experiences, we found that many potential users of a targeting assistant wanted properties of an intelligent data base. In particular, many of the proposed target selection heuristics that arose corresponded to ways to prioritize or filter choices among sets. TASS supported this process by providing an English front end for the TALIE data base. Unlike most data-base query systems, TASS allowed the user to specify new relationships from old ones and to access these with English phrases. Thus, the user could say that the top 10 airfields among those with good weather would be targeted, where "good weather" would itself be a filter function on other data-base items. Each user could define his own domain concepts and selection functions in English.

TASS made two things clear to us. First, users will find the capability to define their own concepts in English extremely desirable. Second, a system like TASS that composes selector-filters in pipeline or hierarchical structures can perform very efficiently. The results of intermediate selections can be cached, and data-driven ramifications of data changes can be propagated directly to the affected user-defined concepcs.

TATR (ROSIE-I Version)

The fourth system we developed was again called TATR and was developed in Rand's initial version of the ROSIE programming language. This current TATR system will be described in the next section; the following paragraphs briefly characterize the lessons we have learned from TATR and ROSIE-I.

This TATR focuses on interdiction and offensive counterair missions in a theater war. It uses knowledge about ground-war and air-ground interactions. It attempts to translate high-level mission objectives into specific targeting recommendations. Like the earlier systems, TATR addresses a narrow part of a big problem, using a limited knowledge-engineering tool. By implementing the knowledge available to us in this area, we have come to believe that an even more focused problem would provide yet a better basis for an intelligent-assistant application.

In terms of technical dimensions, TATR showed many deficiencies in the original ROSIE design. Largely because of the experience derived from applying ROSIE-I to the targeteering problem, Rand has substantially redesigned and reimplemented ROSIE. It is now faster, more versatile, and more terse. In addition, its English syntax is much more natural and more extensive.

SUMMARY

Our initial prototype system developments have had a significant effect on conceptions of (1) what tactical planning requires, (2) what tactical targeteering involves, (3) what parts of targeteering are worth addressing and are technically feasible for knowledge engineering, (4) what capabilities general knowledge-engineering languages should have, and (5) what capabilities a knowledge-engineered tool for the tactical area should possess. These are substantial products, and each feeds into work now under way using ROSIE-II to develop an intelligent assistant for air base attack planning.

IV. TACTICAL AIR TARGETING RECOMMENDER (TATR)--ROSIE-I VERSION

As stated above, our latest version of TATR was developed in the ROSIE-I programming language, supplemented by INTERLISP subroutines for specific mathematical/statistical calculations. It expanded the scope of targeting interest over earlier experiments to include battlefield air interdiction and offensive counterair objectives, added a weaponeering capability, and generally scaled up the target environment, rulesets, and user interface. We feel that most of the essential elements in target selection and weaponeering have been represented adequately to warrant investigation of the technical feasibility of further development. However, we do not claim that the TATR structure and hueristics approach the complex interactions and functions of real targeteers or that this TATR could be considered a true operational assistant as it stands, even if it proved to be technically successful. It would still be only the starting point for evolutionary development.

Ultimately, this version of TATR, like its predecessors, exhibited fatal technical flaws as far as a potential operational system is concerned. However, it provided an important insight into requirements for an improved ROSIE and for further focusing our targeting-assistant research. The following sections address the TATR programming language, ROSIE-I; the TATR program; and its limitations.

TATR PROGRAMMING LANGUAGE--ROSIE-I

ROSIE-I is a programming language designed to support a wide range of knowledge-based programming tasks. Being a direct descendant of

RITA, it is a production rule-oriented language with English-like syntax. It also incorporates a number of improvements suggested by extensive use of RITA by Rand staff members and others.

A program in ROSIE-I consists of a set of production rules, each with an optional antecedent (condition) and a consequent (action). The condition might represent a situation in the real world, such as a buildup of aircraft on an airfield, and the action could be a procedure to follow when that situation is detected. An example of such a rule is given below:

```
Rule 1: IF THERE IS AN AIRFIELD
        WHOSE BOMBER-COUNT IS GREATER THAN 20
        AND WHOSE HELICOPTER-COUNT IS GREATER THAN 30
        THEN USE SELECT-TARGET(THE AIRFIELD);
```

Rules are typically organized into blocks called rulesets. These blocks represent chunks of knowledge or procedures required by the program and are parameterized like function calls in standard programming languages. In TATR, for example, there are rulesets for estimating firepower capacity of an airfield, computing distances between targets, dictating weaponeering policies, weaponeering a target, etc. In the example above, SELECT-TARGET is the name of a ruleset which will choose its parameter, the airfield, as a target for attack.

In addition to rules and rulesets, ROSIE-I programs usually involve a data base, facts about the program's domain of operation. ROSIE-I's data base is a set of objects, such as airfields or helicopters, and attribute-value pairs for each object. For example, an airfield might have 13 bombers, 2 helicopters, and 10 munitions dumps. A data base entry might be created with the following statement:

```
CREATE AN AIRFIELD
        WHOSE BOMBER-COUNT is 13
        AND WHOSE HELICOPTER-COUNT IS 25
        AND WHOSE MUNITIONS-DUMPS IS 10;
```

ROSIE-I represents a great improvement over the RITA programming environment, but the language is still not adequate for the targeting task as we now perceive it. TATR, although it represents a first-cut solution, involves some 300 rules and over 300 data-base objects. Although this is a large program for ROSIE-I, we anticipate that it will be dwarfed by future attempts at more complete solutions to the problem. In anticipation of this, the development of ROSIE-II is well under way, motivated by our experience with ROSIE-I and the targeting task.

THE TATR PROGRAM

Overview

TATR is an interactive program which provides the following information at the specification of the user: preference-ordered target lists; air defense targets associated with those targets; weaponeering options for each target; damage expectancy for specified weapons systems configuration and quantity; and displays of target data. Selectable target sets are airfields, specific targets on an airfield, and ground combat units. Airfield and ground combat unit preferences are based primarily on the perceived threat to friendly ground and air forces. TATR recommends targets to attack on a specified airfield in order to accomplish one or more objectives against that airfield. These objectives are interrupt operations, aircraft attrition, and sortie attrition.

The weaponeering function incorporates rules concerning the commander's policy (such as restrictions on the use of aircraft and munitions and acceptable aircraft attrition), as well as the usual probabilities of arrival and damage. It provides expected damage to a target and the expected attrition to friendly aircraft for each feasible weapon system option, and it sorts those options in accordance with a preference ordering.

TATR's user interface features quasi-natural language communication for selection of the various targeting and weaponeering function options and assists and prompts the user as necessary. All output is in abbreviated textual form.

The following paragraphs review the user interface, the selection of targets using COMPUTE functions, and the weaponeering of targets using COMPUTE OPTIONS functions. Finally, the limitations of this version of TATR are reviewed.

User Interface

TATR's interface with a user is in the form of query and response. TATR initates correspondence with:

```
Welcome to Mod0 -- type ? at any time for options.
Command?
```

Typing "?" will produce the response in Fig. 2, which presents an outline of the command options.

```
Command? ?

Type one of the following:


                COMPUTE AIRFIELDS
                COMPUTE GCUS*
                COMPUTE TARGETS ON AIRFIELD
                COMPUTE TARGETS TO SUPPORT FCPS*
                COMPUTE OPTIONS


                NAME <one of above options>
                LIST <one of above options>
                SAVE <one of above options>


                ADD AIRFIELD
                ADD GCU
                ADD TARGET ON AIRFIELD
                ADD TARGET TO SUPPORT FCPS
                DELETE <one of above options>


                DISPLAY <any target type or FCP>
                UPDATE <any target type or FCP>
                QUIT


                What commands do:
                COMPUTE:  preference orders.
                COMPUTE OPTIONS:  determines weaponeering preferences.
                NAME:  displays names of computed targets in order.
                LIST:  displays computed targets with relevant data.
                SAVE:  same as LIST but puts it on MODO.RESULTS.
                ADD:  adds a target to the saved results list.
                DELETE:  removes a target from saved results.
                DISPLAY:  shows all data for any target.
                UPDATE:  allows you to change target or FCP data.
                QUIT:  gets you out of MODO and into ROSIE.
```

*GCU is ground combat unit(enemy). FCP is friendly critical point, a location used to mark the forward units of friendly ground forces.

Fig. 2--The command options offered by TATR

The command COMPUTE followed by the name of one of the target types, GCUS, AIRFIELDS, TARGETS ON AIRFIELD, and TARGETS TO SUPPORT FCPS, causes TATR to perform target selection functions discussed in the COMPUTE sections below. The command COMPUTE OPTIONS initiates the weaponeering functions.

Once a preference ordering has been determined using one of the COMPUTE commands, the results can be displayed using either the command NAME or LIST. NAME displays only the target identification, while LIST also shows why a target was selected in the listed order (see Figs. 5, 6, and 7 for examples). SAVE stores the results for future access, possibly as the basis of a frag order. LIST OPTIONS, NAME OPTIONS, and SAVE OPTIONS perform the same functions for weaponeering results.

Targets can be added to and deleted from a SAVEd list by the commands ADD TARGET and DELETE TARGET.

DISPLAY displays all information on a specified target contained in the data base. Figure 3 is an example of the displayed data for an airfield and a supply depot.

TATR has a number of automatic informational and prompting responses to user actions. On recieving an instruction that may take time to perform, such as COMPUTE, TATR immediately replies "Scanning airfields..." or "Scanning ground combat units..."; this message remains on the user's terminal screen until the results are displayed. If told to DELETE TARGET ON AIRFIELD, TATR replies "Name or BE number of airfield?" If the user provides an incorrect name or BE number, TATR replies "Ambiguous name, try again" or "No target with that BE number, try again." If the user is correct, TATR asks "Name of target at

```
Command? DISPLAY AIRFIELD
Name or BE-number of AIRFIELD? FALKENBERG
AIRFIELD: FALKENBERG
        BE-NUMBER is 9030
        DESCRIPTION is "FALKENBERG "
        TARGETTYPE is AIRFIELD
        LOCATION is (5132 1313)
        CEILING-HEIGHT is 14000
        VISIBILITY is 12
        SHELTERS is 40
        OPEN-REVETMENTS is 30
        RUNWAYS is 1
        RUNWAY-LENGTH is 8400
        RUNWAY-WIDTH is 200
        RUNWAY-SURFACE is HARD
        CUTS-REQUIRED is 2
        NORMAL-MAINTENANCE-FACTOR is T
        MAINTENANCE-FACILITY is HARD
        POL-LOCATIONS-UNDERGROUND is 4
        LARGEST-UNDERGROUND-POL-STORAGE is 35
        MUNITIONS-LOCATIONS is 0
        LARGEST-MUNITIONS-STORAGE is 0
        AIRCRAFT is (BADGER FISHBED FLOGGER)
        BADGER is 72
        FISHBED is 36
        FLOGGER is 24
        CLOSEST-FCP is F12
        DISTANCE-TO-LANDOP is 174
        THREAT-LEVEL is 1728
        AIRFIELD-OBJECTIVES is (SORTIE-ATTRITION)
        AIR-DEFENSE-TARGETS is ("9876-08041")

Command? DISPLAY SUPPLY-DEPOT
Name or BE-number of SUPPLY-DEPOT? KBELY
SUPPLY-DEPOT: KBELY SUPPLY DEPOT
        BE-NUMBER is 1006
        DESCRIPTION is "KBELY SUPPLY DEPOT "
        TARGETTYPE is SUPPLY-DEPOT
        LOCATION is (5007 1434)
        MUNITIONS-LOCATIONS is 25
        LARGEST-MUNITIONS-STORAGE is 4
        POL-DRUMS is T
        TANKS is 25
        TRUCKS is 40
```

Fig. 3--An example of data provided with the command DISPLAY

airfield?" If the user specifies the target, TATR removes the target and says, "Removed from list." If the user gives an incorrect reply, TATR responds with "Not in list!" or "Type one of the following: Runway, Aircraft, etc. (through list of target types)."

This interface is very basic. Standardization of instructions and response terms in the next version of TATR will simplify the user's job and reduce learning time.

Compute Airfields

TATR's job is to select air interdiction targets that offer the the greatest direct threat to friendly ground and air forces. When TATR is instructed to COMPUTE AIRFIELDS, it selects enemy airfields according to the following heuristic:

> Compute a list of targets ordered by the number of bombs per day deliverable from the airfield to the land operation by Ground Force Attack Aircraft (GFAA) and to friendly airfields by Offensive-Counter-Air (OCA) aircraft. Airfields which can't deliver bombs for either purpose are excluded from the list.

To make the computation, TATR uses rulesets containing necessary information in rules. With a ruleset MISSIONS-PER-DAY (Fig. 4), it determines the number of missions each type of aircraft in the data base can fly and any distance limitations that may apply. According to the rules, bombers can fly one mission per day and helicopter gunships three missions per day if the distance to the target exceeds 60 miles, or five missions per day if the distance is less than 60 miles.

Other rulesets, BOMB-CAPACITY, BOMBS-PER-DAY-TO-LANDOP, OCA-BOMBS-PER-DAY, and DISTANCE-TO-LANDOP, provide the additional rules TATR

```
     Create a ruleset named MISSIONS-PER-DAY
        whose arguments is (aircraft, distance-to-target)
        and load rules into MISSIONS-PER-DAY:

MPD1:   If aircraft is [ a BOMBER ]
                       in (BEAGLE, BADGER, BREWER)
             then return 1:

MPD2:   If aircraft is [ a GROUND-FORCE-ATTACK-AIRCRAFT ]
                       in (FISHBED-D, FISHBED-F, FLOGGER-D, FITTER)
             then return 1.5:

MPD3:   If aircraft is [ an OFFENSIVE-COUNTER-AIR-AIRCRAFT
             and aircraft is not a BOMBER ]
                       in (FLOGGER-D, FITTER)
             then return 1.5:

MPD4:   If aircraft is [ a HELO-GUNSHIP ]
                       in (HOOK, HIP, HIND, HOUND, HOPLITE)
             and distance-to-target is less than or equal to
             60 then return 5:

MPD5:   If aircraft is [ a HELO-GUNSHIP ]
             in (HOOK, HIP, HIND, HOUND, HOPLITE)
             and distance-to-target is greater than 60
             then return 3:

MPD6:   Return 0:

Deactivate MISSIONS-PER-DAY;
```

Fig. 4--Ruleset MISSIONS-PER-DAY in COMPUTE AIRFIELDS

needs to compute the threat level in bombs per day. In BOMB-CAPACITY, the number of bombs for each aircraft is related to a distance-to-target (airfield) for OCA aircraft and a distance-to-landop for GFAA aircraft. Bombers are unconstrained in range and are always considered to carry the same number of bombs. The GFAA aircraft have a range limit of 250 nautical miles, and they carry more bombs if the distance to target is less than 125 nautical miles. Helicopter gunships are considered a threat only if they are within 100 nautical miles of land operations; however, their bomb capacities (bomb-load equivalents) remain constant at all lesser distances and are related to the size of the craft.

To compute the distances from enemy airfields to friendly ground forces or airfields, TATR refers to the current location of 12 "friendly critical points" (FCPs) along the battle line whose locations are maintained in the data base. For distance to a friendly ground unit or units, TATR computes the distance from the enemy airfield to either the closest FCP or an FCP specified by the user. For the distance to friendly airfields (whose locations are not in TATR's data base), TATR uses the distance to the nearest FCP and adds 100 nautical miles. The 100 nautical miles represents an estimate of the average distance from the battle line (FCPs) to all friendly airfields. With this formula, the airfield-to-airfield distance varies with the movement of the battle line. The assumption is made that the operations at forward friendly airfields move as the battle line moves, thus moving the average of the distances with the battle line.

While not treated in a separate ruleset, the mission capability of each type of enemy aircraft is a significant factor in establishing an airfield's level of threat and is recognized in other rulesets. Bombers are considered as OCA aircraft only. Their threat is against friendly airfields and not front-line ground forces. GFAA threaten only the front-line ground forces. The Fishbed aircraft is used as an example of a single-mission GFAA. A third group of enemy aircraft types (Flogger and Fitter) are considered to perform both missions and threaten both airfields and ground forces. The bombs per day of bombers threatening airfields only and of Fishbeds threatening ground forces only are counted once in computing an airfield's threat. The dual-mission Fitters' and Floggers' bombs per day are counted against each target type to reflect their versatility and capability.

Since TATR is concerned only with interdiction targets, it does not consider airfields within 25 kilometers (17 nautical miles) of any FCP, since these are close-air-support targets.

While performing the COMPUTE function, TATR displays "Scanning airfields." When TATR is completed, the user can request a simple listing of targets in rank order by the command NAME AIRFIELDS, as shown in Fig. 5, or a listing of ordered targets including information on the

```
Command? NAME AIRFIELDS

Airfields selected in order of importance:

    9876-09006    DRESDEN
    9876-09015    MIROW
    9876-09030    FALKENBERG
    9876-09023    TEMPLIN 1
    9876-09026    PRAGUE KBELY
    9876-09017    PARCHIM
    9876-09028    MARXWALDE
    9876-09022    STENDAL
    9876-09025    CESKE BUDEJOVICE
    9876-09003    BRANDENBURG PRIEST
    9876-09031    ALT-LONNEWITZ
    9876-09032    FINSTERWALDE
    9876-09001    BARTH
    9876-09009    HAINA
    9876-09016    NEURUPPIN
    9876-09018    PEENEMUNDE
    9876-09011    KARL MARX STADT
    9876-09010    JUTERBORG
```

Fig. 5--TATR's targets recommended with the command NAME AIRFIELDS targets by the command LIST AIRFIELDS, as in Fig. 6. With LIST AIRFIELDS, the objective of an attack, which TATR determines, is shown with the targets. Also provided is a list of the SAM sites that cover

```
Command? LIST AIRFIELDS


Selected airfields in order of importance:

DRESDEN
BE number: 9876-09006
Location: 5108N 1346E
Distance to landop: 105.4nm
Bombs per day: 2880
Objectives against airfield: (SORTIE-ATTRITION)
Defending SAM sites: (9876-08008)

MIROW
BE number: 9876-09015
Location: 5318N 1245E
Distance to landop: 84.94nm
Bombs per day: 1779
Objectives against airfield: (INTERRUPT-OPERATIONS SORTIE-ATTRITION)
Defending SAM sites: NONE

FALKENBERG
BE number: 9876-09030
Location: 5132N 1313E
Distance to landop: 107.88nm
Bombs per day: 1728
Objectives against airfield: (SORTIE-ATTRITION)
Defending SAM sites: (9876-08041)

TEMPLIN 1
BE number: 9876-09023
Location: 5302N 1333E
Distance to landop: 115.32nm
Bombs per day: 1692
Objectives against airfield: (SORTIE-ATTRITION)
Defending SAM sites: NONE
Objectives against airfield:  (SORTIE-ATTRITION)
Defending SAM sites: NONE
```

Fig. 6--TATR's targets (partial list), including details, recommended with the command LIST AIRFIELDS each target. A ruleset IN-RANGE has the rules defining the range of each type of SAM in the environment.

Compute Targets on Airfield

When instructed to COMPUTE TARGETS ON AIRFIELD, TATR generates a preference list of targets located on an airfield that satisfy one or more attack objectives specified by the user. Targets of interest are those items that are critical to the enemy's conduct of air operations, e.g., aircraft, maintenance, munitions storage, POL storage above ground, POL storage underground, and takeoff and landing surfaces (runways). The choices of objectives are "interrupt operations," "aircraft attrition," and "sortie attrition."

TATR's rule for interrupt operations is simply to close the takeoff and landing surfaces. It therefore recommends "runways" as the target on the airfield. It will also recommend "runways" under the other two attack objectives if there are GFAA on the airfield and the airfield is within 100 nautical miles of the battle line.

With an objective of aircraft attrition, TATR may recommend any of three types of target in preference order: aircraft, in the open or uncovered; covered revetments; and shelters, assumed to hold two aircraft each. The data base lists the number of each type of aircraft, the number of revetments, and the number of shelters. TATR first chooses aircraft, if any are uncovered; then covered revetments, if any exist and half of the aircraft are not uncovered; and then shelters, if the sum of aircraft in shelters is more than half of the total aircraft.

When the attack objective is sortie attrition, TATR chooses its recommendations from among five of the six types of targets: aircraft, maintenance, POL storage above ground, POL storage underground, and munitions storage. TATR selects aircraft as top preference when any are uncovered, and maintenance as second preference whenever it is "soft." TATR then considers POL storage; it evaluates the data base information on the capacity of storage above ground and the size of the largest underground storage (both expressed in percent of total POL). It chooses POL storage above ground when that capacity is more than 60 percent of the total and the capacity of storage underground is less than 30 percent. It includes POL storage underground when the largest single underground storage is greater than 30 percent of the total POL. TATR then includes maintenance if it is "hard," rather than "soft." And finally, TATR chooses munitions storage if the largest munitions storage area on the airfield is greater than 50 percent of the total munitions storage.

The result of TATR's computation with the command NAME TARGETS ON AIRFIELD is displayed in Fig. 7. The user's attack objective is provided as in LIST AIRFIELDS.

Compute GCUS

When TATR is given the command COMPUTE GCUS, it seeks the enemy ground combat units that may be the first to attack friendly ground forces. It reviews each GCU (tank division and mechanized infantry division) in the data base, determines its location and the time since it last moved, and orders the GCUs according to threat rules. GCUs

```
Command? NAME TARGETS ON AIRFIELD

Name or BE Number of Airfield?  DRESDEN

Airfield: 9876-09006    DRESDEN
Objectives: (SORTIE-ATTRITION)
Targets on airfield in order of importance:

    AIRCRAFT
    MAINTENANCE
    MUNITIONS-STORAGE
    COVERED-REVETMENTS

(Other examples of output from NAME TARGETS ON AIRFIELD)

Airfield: 9876-09003    BRANDENBURG/PRIEST
Objectives: (INTERRUPT-OPERATIONS SORTIE-ATTRITION)
Targets of airfield in order of importance:

    RUNWAY
    AIRCRAFT
    MAINTENANCE
    POL-STORAGE-ABOVEGROUND

Airfield: 9876-09010    JUTERBORG
Objectives: (SORTIE-ATTRITION)
Targets on airfield in order of importance:

    AIRCRAFT
    MAINTENANCE
    POL-STORAGE
    MUNITIONS-STORAGE

Airfield: 9876-09018    PEENEMUMDE
Objectives: (AIRCRAFT-ATTRITION)
Targets on airfield in order of importance:

    AIRCRAFT
    COVERED-REVETMENTS
    SHELTERS
```

Fig. 7--Types of targets on Dresden Airfield recommended with the command NAME TARGETS ON AIRFIELD

within one night's move, 50 kilometers, are ordered by distance from the closest FCP. GCUs beyond 50 kilometers are ordered by the number of nights required for them to move to the land operation, up to a maximum of three. Those that tie under the above two rules are considered more threatening with increasing time since their last observed move. GCUSORT is a LISP function which orders the GCUs by the above criteria. GCUs less than 25 kilometers distant are considered to be close-air-support targets, so they are not included on the list.

Figure 8 shows a portion of the COMPUTE-GCUS ruleset. Hyphenated words in capital letters are other rulesets which execute the procedures described above.

Compute Targets to Support FCPS

When TATR is given the command COMPUTE TARGETS-TO-SUPPORT-FCPS, it determines the GCUs and airfields that directly threaten FCPs specified by a user.

For GCUs, the same type of rules are used in this instruction as in COMPUTE GCUs, except that distances are measured to the specified FCPs rather than to the closest FCP, and only the GCUs that lie closer to the specified FCPs than all other FCPs are selected for ordering.

The airfield threat is measured by the same type of rules as in COMPUTE AIRFIELDS, except that only airfields with GFAA and helicopter gunships are considered, and the threat level is represented by the number of bombs deliverable per day by these aircraft only in their GFAA role. Airfields that cannot deliver bombs to any of the specified FCPs are excluded from the list, as are airfields that are close-air-support targets within 25 kilometers of the specified FCPs.

```
        Create a ruleset named COMPUTE-GCUS
                whose local-varibles is
                    (ordered-set)
        and load rules into COMPUTE-GCUS;

        [ Computes an ordered list of GCU targets. ]

GCU1:   If there is a GCU
        whose distance-to-landop is not known
        then for each case
        set distance-to-landop of the GCU to
                DISTANCE-TO-LANDOP(the GCU);

GCU2:   If there is a GCU
                whose distance-in-days is not known
                then for each case
                set distance-in-days of the GCU to
                        GCU-DISTANCE-IN-DAYS(distance-to-landop of the GCU);

GCU3:   If there is a GCU
                whose time-since-last-move is not known
                then for each case
                set time-since-last-move of the GCU to
                        TIME-SINCE-LAST-MOVE(the GCU);
                                .
                                .
                                .
                                .
GCU7:   Set gcus of results to GCUSORT(ordered-set) and return;

        Deactivate COMPUTE-GCUS
```

Fig. 8--Ruleset COMPUTE-GCUS

The display of targets received after TATR's computations is shown in Fig. 9. Note that the measurement of threat, bombs per day for airfield aircraft, and distance (days) to FCPs and hours since last move for GCUs is included with each target.

```
Targets in support of FCPs: (F12)
Airfields selected in order of importance:

STENDAL
BE number: 9876-09022
Location: 5238N 1149E
Distance to specified FCP(fcps): 42.78nm
Bombs per day to specified FCP(s): 870
Objectives against airfield: (SORTIE-ATTRITION)
Defending SAM sites: (9876-08023 9876-08024 9876-08025)

MIROW
BE number: 9876-09015
Location: 5318N 1245E
Distance to specified FCP(fcps): 101.68nm
Bombs per day to specified FCP(s): 864
Objectives against airfield: (SORTIE-ATTRITION)
Defending SAM sites: NONE

TEMPLIN 1
BE number: 9876-09023
Location: 5302N 1333E
Distance to specified FCP(fcps): 119.04nm
Bombs per day to specified FCP(s): 846
Objectives against airfield: (SORTIE-ATTRITION)
Defending SAM sites: NONE

GCUs selected in order of importance:

16TH MECHANIZED RIFLE DIVISION
BE number: 9876-13015
Location: 5237N 1110E
Distance to specified FCP(s): 24.18nm (1 days)
Hours since last move: 74

15TH MECHANIZED RIFLE DIVISION
BE number: 9876-13014
Location: 5215N 1204E
Distance to specified FCP(s): 45.88nm (2 days)
Hours since last move: 74

12TH TANK DIVISION
BE number: 9876-12011
Location: 5130N 1202E
Distance to specified FCP(s): 70.06nm (3 days)
Hours since last move: 27
```

Fig. 9--TATR's targets (partial list) recommended with the command NAME TARGETS TO SUPPORT FCPS

Weaponeering

The weaponeering function is initiated with the command COMPUTE OPTIONS. This function provides all combinations of weapon system and tactics that can achieve a specified desired damage expectancy (DE) against a target. The target may be any of twelve types: GCUs, airfields, supply depots, highway bridges, highway junctions, SAM sites, military headquarters, army barracks, power plants, radar sites, railroad bridges, and railroad yards. When instructed to COMPUTE OPTIONS, TATR asks for a target name or BE number.

If the target is an airfield, TATR first selects and orders the targets on the airfield (aircraft, maintenance, POL storage, munitions storage, and runways) according to the rules in COMPUTE TARGETS-ON-AIRFIELD (discussed above) and assigns an attack objective of "sortie attrition." If the airfield contains GFAA and is within 100 nautical miles of the battle line, the "interrupt operations" objective is also assigned.

For all types of targets, TATR then determines target characteristics and number of aim points (for example, the number of cuts required to close a runway or the number of POL storage locations). If it is not given a desired DE by the user, TATR will use a default DE set by commander policy. If there are multiple aim points for a target (e.g., six POL storage tanks), TATR computes the number of aircraft for each aim point such that the DE achieved against each aim point is equal to or greater than the desired DE for the target (POL storage).

With the basic target information and desired DE, TATR then begins the many steps of WEAPONEER-TARGET to develop a list of recommended weapon system/target combinations. It utilizes rulesets that incorporate heuristic means for shortening otherwise long and time-consuming processes and for expressing policy, criteria, and limitations. These rulesets are briefly described below.

SUGGEST-MUNITIONS, one of the first rulesets employed, directs TATR's weaponeering efforts to the three munitions that by previous weaponeering show themselves to be the most effective under various delivery conditions.

SUGGEST-SYSTEM narrows the search for an appropriate weapon system in TATR's review of the ruleset WEAPONS-SYSTEM, by which TATR knows all feasible aircraft type/munitions combinations.

MUNITIONS-POLICY prohibits the use of certain munitions on certain types of targets, for instance, CBU on bridges or hard maintenance.

AIRCRAFT-POLICY guides aircraft use, such as primarily employing F111's for the most difficult targets and limiting the distance to target for A-10s to enhance survival and to enable ready diversion to close-air-support missions.

COMBAT-RADIUS-CHECK verifies preplanned distance limits for aircraft types.

WEATHER-LIMITS provides primarily for rules concerning a commander's weather go/no-go decisions for missions. TATR's current rules eliminate any weapon system/munition combinations using delivery tactics that require a ceiling or visibility exceeding the target weather.

WEAPONEER-PACKAGE weaponeers the resulting combinations of weapon systems and targets. A probability of arrival is determined using a probability of survival (Ps) enroute and at the target. The Ps at the target is dependent on the tactics used by the aircraft (low, high, or level) and the air defense associated with the target (light or heavy). If not specified in the data base, the defense is assumed to be heavy. Using the probability of arrival at the target (Pa) and the probability of kill (Pk) of the munitions against the target type, this ruleset determines the number of aircraft required to achieve the desired DE. It then rounds the number of aircraft up to an integer, computes the resulting "actual" DE, and adjusts the number of aircraft upward to account for attrition. (For additional information, see p. 15.)

Recognizing a possibility for an operational limitation on the number of aircraft in an attacking force, TATR provides a ruleset REDUCTION-REQUIRED to exercise a mission size policy and a limit on aircraft carrying precision munitions. Any reduced package is re-weaponeered to determine actual DE.

TATR has two final weaponeering checks: MISSION-POLICY provides for any long-term or daily policy variables that may limit selection of any weapon system/target combination, such as political limitations on targets or geographical areas, aircraft or munitions restrictions, and resource value policy. As an example of resource value policy, TATR requires an actual DE of at least .3. A second check is ACCEPTABLE-ATTRITION, in which TATR compares the expected attrition to a set of rules that establish the maximum loss for a given DE against a given number of targets or size of target. As an example, if the target is aircraft at an airfield, attrition is unacceptable unless at least 12 enemy aircraft are destroyed for each attacking aircraft lost.

ORDER-PACKAGES, the final step, orders the list of strike packages by rules of preference and displays the reordered list. TATR first considers the combinations in three groups established by DE. Within the group of DEs greater than .69, it orders by lowest attrition and by delivery tactic. Low and level deliveries are preferred over high, because the probability of having delivery weather minimums and a successful mission as well is greater for low and level, and Ps is also greater. TATR orders the groups of targets with lesser DEs in the same way. If it can find no acceptable options for a target, TATR informs the user of the rejected options and gives the reason for rejection with "FAILS <ruleset name>."

In the display of final weaponeering results, TATR provides the weapon system options it recommends for each type of target on the airfield. The data include the aircraft attrition and expected damage, so that a user can check TATR's recommendations. When the number of aircraft required to achieve the desired DE of .69 exceeds the maximum established force size, TATR displays only the maximum number of aircraft and the term "(reduced)." In Fig. 10, Dresden airfield is the target and LIST TARGETS ON the command. TATR has recommended as targets aircraft, maintenance, and munitions storage, in that order. Figure 11 is an example of weaponeering against a storage facility and a bridge.

LIMITATIONS

Although TATR now has a significant capability to perform certain types of tactical air targeting functions, it also has severe limitations which prohibit its consideration as a basis for a realistic

```
Weaponeering options against GADEBUSCH STORAGE FACILITY in recommended
order of priority:

WAREHOUSES
    Aircraft: 11 F-4X
    Munitions: (MK82 MK84)
    SCL: 1
    Tactic: Low
    Probable aircraft attrition: 1.32
    Damage expected: .69

    Aircraft: 8 A-10X
    Munitions: (MK82 CBU)
    SCL: 2
    Tactic: Low
    Probable aircraft attrition: .63
    Damage expected: .73

    Aircraft: 16 F-4X  (reduced)
    Munitions: (MK82 MK84)
    SCL: 1
    Tactic: High
    Probable aircraft attrition: 2.55
    Damage expected: .64

Weaponeering options against OEBISFELDE HIGHWAY BRIDGES in recommended
order of priority:

    Aircraft: 2 F-111X
    Munitions: (LGB)
    SCL: 2
    Tactic: Level
    Probable aircraft attrition: .19
    Damage expected: .88

    Aircraft: 2 F-4X
    Munitions: (LGB)
    SCL: 4
    Tactic: High
    Probable aircraft attrition: .31
    Damage expected: .87

    Aircraft: 8 F-111X
    Munitions: (MK84)
    SCL: 1
    Tactic: Level
    Probable aircraft attrition: .79
    Damage expected: .7
```

Fig. 11--Weaponeering against a storage facility and bridges

Weaponeering options against GADEBUSCH STORAGE FACILITY in recommended order of priority:

WAREHOUSES

Aircraft: 11 F-4X
Munitions: (MK82 MK84)
SCL: 1
Tactic: Low
Probable aircraft attrition: 1.32
Damage expected: .69

Aircraft: 8 A-10X
Munitions: (MK82 CBU)
SCL: 2
Tactic: Low
Probable aircraft attrition: .63
Damage expected: .73

Aircraft: 16 F-4X (reduced)
Munitions: (MK82 MK84)
SCL: 1
Tactic: High
Probable aircraft attrition: 2.55
Damage expected: .64

Weaponeering options against OEBISFELDE HIGHWAY BRIDGES in recommended order of priority:

Aircraft: 2 F-111X
Munitions: (LGB)
SCL: 2
Tactic: Level
Probable aircraft attrition: .19
Damage expected: .88

Aircraft: 2 F-4X
Munitions: (LGB)
SCL: 4
Tactic: High
Probable aircraft attrition: .31
Damage expected: .87

Aircraft: 8 F-111X
Munitions: (MK84)
SCL: 1
Tactic: Level
Probable aircraft attrition: .79
Damage expected: .7

Fig. 11--Weaponeering against a storage facility and bridges operational system. For one thing, it is very slow, often requiring as much as 20 minutes to calculate targets in support of specific friendly critical points. Similar times are experienced on other calculations as well. While some of the times could be reduced by more pre-calculation, only limited pre-calculation is possible in a dynamic operational environment. Another major problem is the expanding core requirement during run time, which in our experience soon exceeded the address space in the DEC 2060. This forced us to split TATR into two programs, one for target recommending and one for weaponeering.

These deficiencies constrained TATR's utility as a vehicle for interfacing with Air Force targeteers and caused us abandon further work with this version.

V. FUTURE RESEARCH DIRECTION

Based on our experience with the development of the prototype systems described above, we are now directing our efforts toward building a third TATR prototype targeting assistant using ROSIE-II. We expect that the improved capability of ROSIE-II will give us sufficiently greater calculation speeds and program scope to enable us to have on-line interaction between the targeting community and an operationally interesting problem.

This TATR will address selection of enemy airfields, targets at those airfields, and weapons to attack them. We chose this focus after extensive interactions with Air Force targeteers, in whose judgment counterair targeting is sufficiently important and difficult to benefi+ from an automated aid. Furthermore, counterair targeting has been given extensive thought and commands much interest in the targeting community; hence, a richer source of knowledge was available to us on this problem than on other targeting areas.

Some of the features we intend to investigate in this TATR are on-line (Air Force) user interaction, on-line user program modification, extended data base, assisted data base maintenance, and modest explanation capability. Processing capabilities will include preference ordering of airfields within and among operational capabilities (operational capabilities of interest are air defense, counterair, and ground force attack), automatic development of strike packages (groupings of attack and support aircraft for use against a target or target set), enemy airfield post-strike recovery estimation, and cost/benefit achievements.

REFERENCES

1. Gorlin, Daniel M., Frederick Hayes-Roth, Stanley Rosenschein, Henry Sowizral, and D. A. Waterman, Programming in ROSIE: An Introduction by Means of Example, The Rand Corporation, N-1646-ARPA (forthcoming).

2. ----, The ROSIE Language Reference Manual, The Rand Corporation, N-1647-ARPA (forthcoming).

3. ----, Rationale and Motivation for ROSIE, The Rand Corporation, N-1648-ARPA (forthcoming).